International Journal of Computer Networking,
Wireless and Mobile Communications (IJCNWMC)
ISSN(P): 2250-1568; ISSN(E): 2278-9448
Vol. 5, Issue 4, Aug 2015, 1-8

TRANS STELLAR
• Journal Publications • Research Consultancy

DESIGN APPROACH OF 64-POINT RECONFIGURABLE FFT/IFFT PROCESSOR FOR ASIP ARCHITECTURE

L. P. THAKARE & A. Y. DESHMUKH

[1]Research Scholar, Department of Electronics Engineering, G. H. Raisoni College of Engineering, Nagpur, Maharashtra, India

[2]Professor, Department of Electronics Engineering, G. H. Raisoni College of Engineering, Nagpur, Maharashtra, India

ABSTRACT

In this paper for ASIP, a novel architecture of reconfigurable FFT is proposed. For the singed decimal number, the proposed design implemented for a reconfigurable 64 point FFT processor. Reconfigurable FFT processor is designed for 2, 4, 8, 16, 32 and 64 point FFT for the of integer k which incorporates a high speed ASIP. The part of OFDMA system is Alterable point FFT processor so there is a need of reconfigurable FFT processor. Proposed design is implemented on Xilinx 9.1i for DIF-FFT algorithm for 272.769 MHz clock frequency.

KEYWORDS*:* FFT, DIF-FFT, OFDMA

I. INTRODUCTION

The core part of OFDMA system is FFT module defined by IEEE STD 802.16-2005. FFT module is physical layer of OFDMA which can be used in the FFT points: 2048, 1024, 512, 128 64 points. The design about variable point FFT processor is just based on FFT module in OFDMA system application. Hence this paper uses a reconfigurable FFT design of ASIP in OFDMA system.

The Fast Fourier Transform [FFT] and Inverse Fast Fourier Transform [IFFT] are the two important key points in the OFDM system. FFT is a fast way to calculate the Discrete Fourier Transform [DFT], which transforms data from time domain to frequency domain where as IFFT transforms data from frequency domain to time domain. The hardware implementation of FFT/IFFT approaches is a challenging issue where the digital signal processors (DSPs) and field programmable gate array (FPGA) chips are two considering designing environments for implementing different schemes of FFT approaches. Recently, the FPGA technology is used as due to fast progress in very large scale integration (VLSI) technology.

A variety of ASIP implementations have been presented for FFT algorithms [3] used 2D FFT Algorithm to design the FFT Processor. According to the idea of two-dimensional Fourier algorithm, i.e. N = 128, N1 = 2, N2 = 64. From 128 = 2 * 64. When achieve 128-point FFT, firstly the data is arranged in 64 lines and 2 rows, secondly the input data will transform the 64 points FFT, then the result multiplies twiddle factor, Thirdly, the result decimates to 2 points FFT. For the classic Cooley-Tukey FFT (CT-FFT), different ASIP implementations have been presented. In [4] instruction capable of calculating a whole butterfly operation is implemented and the computation resources are distributed into three execution stages to reduce the clock cycle. However, long pipelines consume more energy, and the speed from one integrated

butterfly computation is still not enough to meet the high throughput requirement of the demanding IEEE 802.15.3a UWB communication standard. In [5] Hardware Extension is used for computations composed by four parallel butterfly units, with a separate custom register file (CRF) to store all the intermediate data of the FFT computations in each epoch, and an on-chip ROM for the intra-epoch coefficients. An address changing logic (AC) is added in the decoder to give the right data address and coefficient address. It utilizes parallelism in the data path for performance improvement. A vectorized Ultra-Long Instruction Word (ULIW) approach is introduced in which performs eight radix-2 butterfly operations in parallel, at the cost of high gate count and power dissipation for the wide instruction length of 619 bits. The Xtensa ASIP design for FFT adds a set of special instructions for computation operations.

Therefore, after analyzed and compared the various FFT algorithms, Decimation-In-Frequency FFT algorithm have been chosen. In order to ensure precision, the floating-point system used in the design.

The paper presents a variable point FFT processor of a pipeline structure which is reconfigurable. The article is structured as follows. Fourier Transforms which is chosen in this design are illustrated in Section II. In Section III, the novel design of ASIP using reconfigurable FFT is proposed. In Section IV, the implementation and simulation is detailed. At last the conclusion and results are discussed.

II. FAST FOURIER TRANSFORM

Fast Fourier Transform(FFT) is defined by the formula:

$$X_k = \sum_{n=0}^{N-1} x(n) W_N^{nk} \quad k= 0,1,\ldots N-1 \quad (1)$$

In equation (1) N is transform length, $W_N^{nk} = e^{-j2\pi/N}$ is the Twiddle factor.

The conventional method of FFT calculation involves N^2 complex multiplication and N (N-1) complex additions.

The radix -2Cooley-Tukey algorithm performs the same computation involving (N/2) log2N complex additions. Algorithms is divided into frequency based DIF- FFT. Figure 1 shows Butterfly representation of signal flow graph for 8 point FFT.

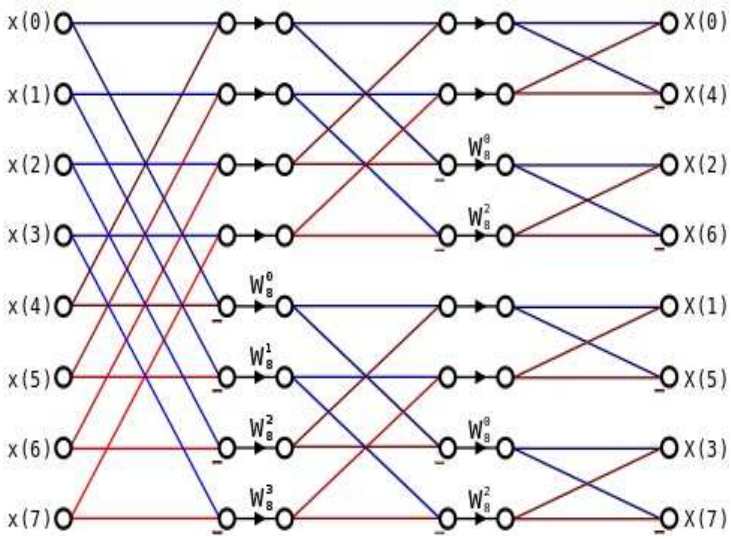

Figure 1: Butterfly Signal Flow Graph of 8 Point FFT

The basic idea of this algorithm is that the N point FFT is segmented into smaller unit's upto two points FFT. Here FFT is decimated in frequency. The 8 point FFT requires three stages of Butterfly computations. Result of first stage butterfly values is multiplied with Twiddle factors Twiddle Factors are considered as Full Twiddle values which are multiplied in the equations. The result will be given to second Butterfly stage. Same type of computations will be done till the structure reduces to 2 point FFT. This will reduce the multiplications using adder and shifter. This needs to convert ternary sum to 3x mantissa and exponent part which takes more multiplications. The proposed method reduces time and computations required. Table 1 and Table 2 [2] gives the value for Twiddle factor multiplier.

Table 1: Cos Twiddle

Cos(2πj/N)	Full Twiddle
1	1
0.923819	9
0.706883	7
0.382245	4
-0.00063	0
-0.38341	-4
-0.70778	-7
-0.9243	-9

Table 2: Sin Twiddle

Sin(2πj/N)	Full Twiddle
1	1
0.382829	4
0.70733	7
0.924061	9
1	1
0.923577	9
0.381661	4
-0.00126	0

III. DESIGN OF ASIP

The core part of ASIP is Reconfigurable FFT module which can be used in various FFT points: 128 points, 64 points, 32 points etc. The design of the select module is based on input constant value given to the constant K to select the required points FFT: 64 points, 32 points, 16 points etc. The algorithm is compiled by defying the value of K. The architecture of the processor was validated and modeled in VHDL language and functionally verified by Xilinx 9.1i software

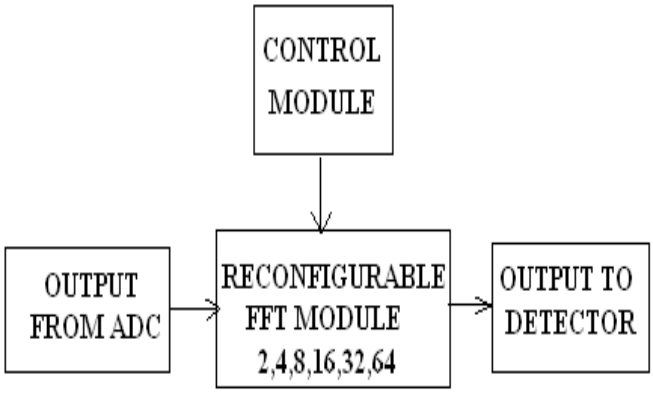

Figure 2: Architecture of ASIP in OFDMA

IV. IMPLEMENTATION AND SIMULATION

At 3.66ns clock cycles, the proposed design was synthesized. For the value of K: 8 the RTL diagrams for Reconfigurable FFT module is shown which is verified according to other efficient ways of FFT calculations.

RTL Schematics

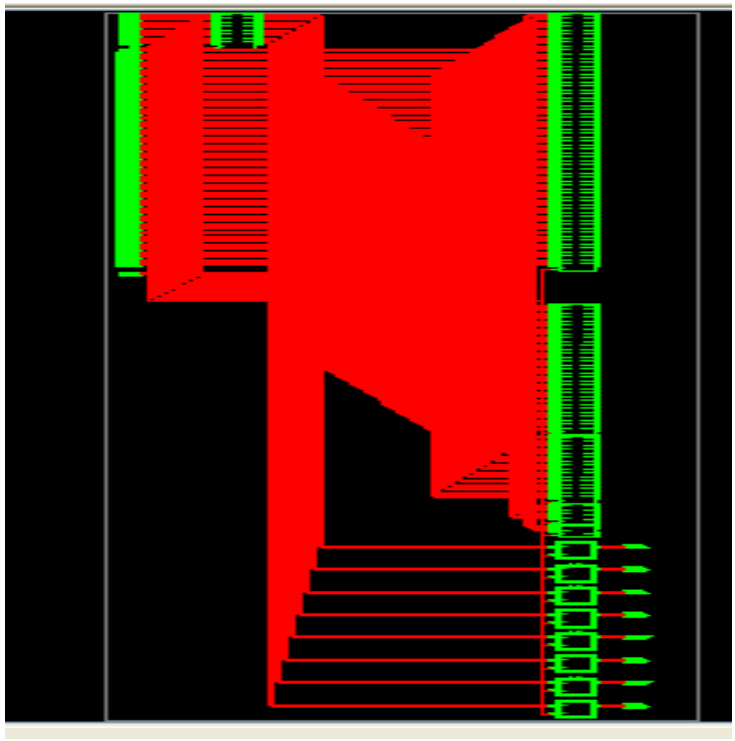

Figure 3: RTL View for 8 Point FFT

Behavior Simulations

The input to the 8 point FFT is of 4 bits. First bit is sign bit. Accordingly signed decimal input from 0 to (-8) can be given to the FFT Processor.

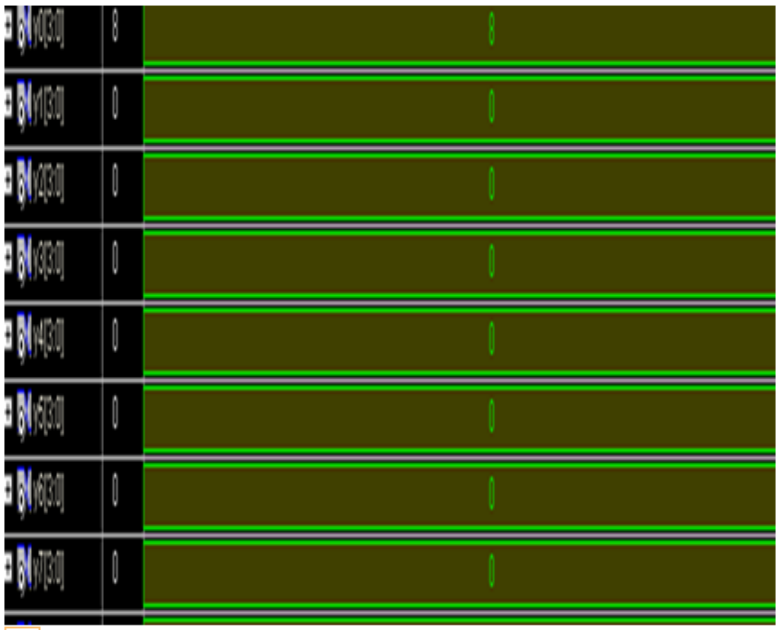

Figure 4: Simulation of 8 Point FFT

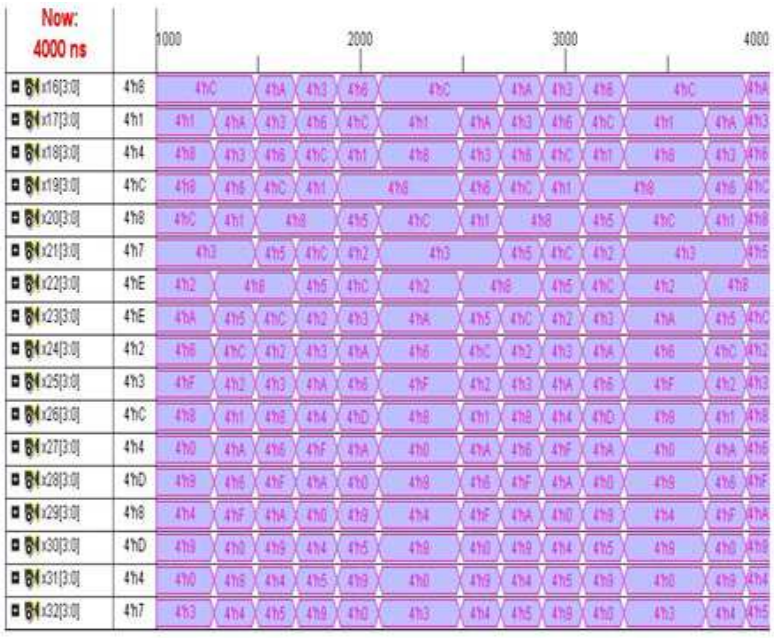

Figure 5: Simulation of 32 Point FFT

Table 3: Statistics for Reconfigurable FFT

Utilization Parameter	2 Points FFT	4 Points FFT	8 Points FFT	16 Points FFT	32 Points FFT	64 Points FFT
4 I/P LUT's %	0	1	2	7	16	46
BONDED IOB%	9	19	37	74	148	296
REGISTERS	7	14	40	258	616	1432
MEMORY(KB)	153068	153068	154092	154092	157740	162860
ADDER/SUB	2	8	24	64	160	384
FLIP FLOPS	16	48	128	320	768	1792

V. RESULTS

Table 3 gives the design statistics for reconfigurable FFT where 513 numbers of Input Output Blocks are utilized which is nearly 68% of the total available.12% of Generic clocks are utilized

VI. CONCLUSIONS

For ASIP, a novel architecture of reconfigurable FFT is proposed and implemented. For the singed decimal number number, the proposed design implemented for a reconfigurable 64 point FFT processor. Reconfigurable FFT processor is designed for 2, 4, 8, 16, 32 and 64 point FFT for the of integer k which incorporates a high speed ASIP and implemented on 272.769 MHz clock frequency. Proposed design is implemented on Xilinx 9.1i for DIF-FFT algorithm. The design can be applied to real-time signal processing system, which completes the main computing modules in the OFDMA system

REFERENCES

1. Hanan M. Hassan, Ahmed F Shalash and Karim Mohd “FPGA Implementation of an ASIP for high throughput DFT/DCT 1D/2D engine” 2012 IEEE.
2. M. Baby latha Rajan and Dr. V. Pushparahavan “64-Point Radix 4 FFT Processing Using DBNS” International Conference on Computer and Information Technology, Bangkok, 2012
3. WANG Xiu-fang, HOU Zhen-long “Design and Implement of FFT Processor for OFDMA System using FPGA” 2011 IEEE.
4. Akshay Sridharan, A Fiji “Low Power Hardware Implementation of High Speed FFT Core” International Conference on Advances in Computer Engineering 2010 IEEE.
5. Xian Guan, Hai Lin and Yunsi Fei. “Design of an Application-specific Instruction Set Processor for High-throughput and Scalable FFT” 2009 IEEE.
6. Preethi sudha Gollamudi, M. Kamaraju Design of high performance IEEE-754 single precision (32 bit) floating point adder using VHDL, IJERT Vol2 issue 7, July 2013.
7. IEEE standard for floating-point arithmetic (IEEE STD 754-2008),revision of IEEE std 754- 1985.august (2008).
8. Karan Gumber,Sharmelee Thangjam “Performance Analysis of Floating Point Adder using VHDL on Reconfigurable Hardware” in International Journal of Computer Applications (0975 – 8887) Volume 46– No.9, May 2012
9. Loucas Louca, Todd A cook and William H. Johnson, “Implementation of IEEE single precision floating point addition and multiplication on FPGAs,”©1996 IEEE.
10. Ali malik, Soek bum ko, “Effective implementation of floating point adder using pipelined LOP in FPGAss,” ©2010 IEEE.
11. Metin Mete, Mustafa Gok, “A multiprecision floating point adder” 2011 IEEE.
12. B. Fagin and C. Renard, “Field Programmable Gate Arrays and Floating Point Arithmetic,” IEEE Transactions on VLSI, vol. 2, no. 3, pp. 365–367, 1994.

13. A. Jaenicke and W. Luk, "Parameterized Floating-Point Arithmetic on FPGAs", Proc. Of IEEE ICASSP, 2001, vol. 2, pp.897-900.
14. N. Shirazi, A. Walters, and P. Athanas, “Quantitative Analysis of Floating Point Arithmetic on FPGA Based Custom Computing Machines,” Proceedings of the IEEE Symposium on FPGAs.
15. N. Quach, N. Takagi, and M.Flynn. On fast IEEE rounding[R]. Technical Report CSL-TR-91-459, Stanford University, January 1991
16. Joseph R. Cavallaro, Predrag Radosavljevic“ASIP Architecture for Future Wireless Systems: Flexibility and Customization” Nokia Corporation, Texas Instruments, Inc 2008 IEEE.
17. A. T. Jacobson, D. N. Truong, and B. M. Baas, “The design of a reconfigurable continuous-flow mixed-radix FFT processor,” in IEEE International Symposium on Circuits and Systems ISCAS.
18. Chidambaram, R. V. Leuken, M. Quax, I. Held, and J. Huisken.”A multistandard FFT processor for wireless system-on-chip implementations “in Proc. Int. Symp. on Circuits & Systems, May 2006.
19. M. Nicola, G. Masera, M. Zamboni, H. Ishebabi, D. Kammler, G. Ascheid, and H. Meyr.” FFT processor: A case study in ASIP development” In IST Mobile Summit, 2005.